UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/632,222	07/31/2003	**Gerard Chauvel**	TI-35452	2081

23494 7590 02/05/2009
TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
MOLL, JESSE R	
ART UNIT	PAPER NUMBER
2181	
NOTIFICATION DATE	DELIVERY MODE
02/05/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

uspto@ti.com

Office Action Summary	**Application No.** 10/632,222	**Applicant(s)** CHAUVEL ET AL.	
	Examiner JESSE R. MOLL	**Art Unit** 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 November 2008*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *11/24/2008*.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 2, 7-12, 25, and 25-31 rejected under 35 U.S.C. 103(a) as being unpatentable over Chennupaty (U.S. Patent No. 6,014,735) in view of Narayan (U.S. Patent No. 6,014,735) in further view of Leijten (US PGPUB No. 2002/0116598).

3. Regarding claim 1, Chennupaty discloses a processor, comprising: instruction storage in which instructions are stored (fig. 3, ref. 310; col. 5 lines 9-16); fetch logic coupled to the instruction storage to fetch instructions from the instruction storage (col. 5 lines 17-18 & 25-26);

Note that since the decoder and escape detector receive instructions, they must be fetched from the instruction buffer.

Decode logic coupled to the fetch logic to decode instructions fetched by the fetch logic (fig. 3, ref. 340; col. 5 lines 24-27); and pre-decode logic associated with the decode logic (fig. 3, refs. 320 and 330; col. 5, lines 17-23); wherein at least some of the instructions comprise a prefix (col. 1, lines 54-57), the pre-decode logic determines whether a subsequent instruction comprise a prefix (col. 5, lines 17-21), in which case the decode logic causes a program counter to skip the prefix and precluding the decode logic from receiving the prefix (see fig. 4, ref. 420 and EN1; col. 5, lines 59-65 regarding enabling the regular on byte decoder 430 only when there is no prefix) and changes behavior of the decode logic during decoding of the subsequent instruction (col. 6, lines 38-53).

Note that the system must keep track of the position of the current instruction. Whatever keeps track of this is considered to be the program counter. Further note that if the first byte is a valid prefix, the first byte is ignored by the decoder, and the second (or third) byte is sent to the appropriate decoding logic in effect causing the program counter (pointer to the current instruction) to skip the prefix bytes and point to the instruction being decoded.

Chennupaty does not expressly disclose that when an instruction is being decoded, the subsequent instruction is being predecoded.

Narayan teaches a method of predecoding instructions in a previous clock cycle (stage) as they are decoded (fig. 3, refs. 54-60, col. 15, lines 14-20).

One advantage for creating multiple cycles for decoding and instruction alignment is to decrease clock time and therefore increase performance (Narayan col. 1, lines 17-37).

It would have been obvious for one of ordinary skill in the art at the time of the invention to combine the inventions of Chennupaty and Narayan by splitting decoding and predecoding (finding the beginning of instructions) into separate clock cycles.

Chennupaty and Narayan do not expressly disclose that incrementing the program counter precludes the decode logic from receiving the prefix.

Leijten teaches incrementing a program counter to skip bytes that should not be executed / decoded (see paragraph [0017]).

At the time of the invention, it would have been obvious for one of ordinary skill in the art to have further modified the invention of Chennupaty and Narayan by precluding the decode logic from receiving the prefix because of incrementing the program counter, As taught by Leijten, to yield predictable results, in order to simplify the design of the processor.

4. Regarding claim 2, Chennupaty/Narayan/Leijten disclose the processor of claim 1, wherein at least some instructions comprise at least one Bytecode (Chennupaty fig. 2, refs. 212, 214, 216, 222, and 224).

Note that the applicant points out in the specification (paragraph 30) that a Bytecode is a byte of instruction.

5. Regarding claim 7, Chennupaty/Narayan/Leijten disclose the processor of claim 1, wherein in parallel with the decode logic decoding the current instruction, the pre-decode logic examines a predetermined number of subsequent bytes (Narayan, col. 6, lines 37-41).

6. Regarding claim 8, Chennupaty/Narayan/Leijten disclose the processor of claim 7, wherein the predetermined number is at least 5 (Narayan, col. 6, lines 50-52).

7. Regarding claim 9, Chennupaty/Narayan/Leijten disclose a method of decoding variable length instructions (Chennupaty col. 3, lines 6-15),

Note that the instructions are variable length because the prefix is a variable length.

Comprising: decoding a current instruction according to a first behavior; while decoding the current instruction, pre-decoding a subsequent instruction to determine if the subsequent instruction includes a predetermined prefix; and if the subsequent instruction includes the predetermined prefix to thereby preclude the decode logic from receiving the prefix, causing a program counter to skip the predetermined prefix and changing the decoding of the subsequent instruction according to a second behavior (see above regarding claim 1).

8. Regarding claim 10, Chennupaty/Narayan/Leijten disclose the method of claim 9, wherein pre-decoding includes examining a predetermined number of bytes following the current instruction (see above regarding claim 7).

9. Regarding claim 11, Chennupaty/Narayan/Leijten disclose the method of claim 10, wherein the predetermined number is at least 5 (see above regarding claim 8).

10. Regarding claim 12, Chennupaty/Narayan/Leijten disclose the method of claim 10, wherein pre-decoding further includes comparing each of the predetermined number of bytes to prefix value (Narayan col. 6, lines 40-50).

Note that the prefix value signifies the start of the instructions therefore the start of the address is found by finding prefix values.

11. Regarding claim 25, Chennupaty/Narayan/Leijten disclose a programmable device, comprising: a register storing a location of a current instruction; a decode logic (fig. 3, ref. 340; col. 5 lines 24-27); and a pre-decode logic coupled to the decode logic (fig. 3, refs. 320 and 330; col. 5, lines 17-23), wherein in parallel, the decode logic decodes the current instruction and the pre-decode logic determines if a subsequent instruction includes a prefix, and wherein if the subsequent instruction comprises the prefix, the program counter skips the prefix of the subsequent instruction thereby precluding the decode logic from receiving the prefix and changes behavior of the decode logic for decoding of the subsequent instruction (see above regarding claim 1).

12. Regarding claim 28, Chennupaty/Narayan/Leijten disclose the programmable device-of claim 25 wherein the current instruction and the subsequent instruction each comprises at least one Bytecode (see above regarding claim 2).

13. Regarding claim 29, Chennupaty/Narayan/Leijten disclose the programmable device of claim 25, wherein the pre-decode logic further determines a predetermined number of subsequent bytes (see above regarding claim 7).

14. Regarding claim 30, Chennupaty/Narayan/Leijten disclose the programmable device of claim 29, wherein the predetermined number is at least 5 (see above regarding claim 8).

15. Regarding claim 31, Chennupaty/Narayan/Leijten disclose the programmable device of claim 25, wherein the register is a program counter.

Claims 3, 4, 13, 15, 16, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chennupaty (U.S. Patent No. 6,014,735) in view of Narayan (U.S. Patent No. 6,014,735) and in further view of Leijten and Google (New bytecodes for "real" Java?).

Regarding claim 3, Chennupaty/Narayan/Leijten discloses the processor of claim 1 (see above regarding claim 1).

Chennupaty/Narayan/Leijten do not disclose the use of a Java impdep instruction as a prefix.

Google discloses the use of an impdep instruction as a prefix for changing instruction sets (Ralf Kraudelt Nov. 6 1998).

Note that the internal opcodes in section 6.2 of the JVM specification are the impdep1 and impdep2 instructions.

One advantage of using the impdep1 and impdep2 instructions would have been to give the ability to access machine dependent instructions and access machine dependent I/O (Google; Ralf Kraudelt Nov. 6 1998).

In light of this advantage, it would have been obvious to a person of ordinary skill in the art at the time of the invention to implement the Java impdep instructions in the system of Chennupaty/Narayan/Leijten in order to switch between a native machine language and Java. These instructions would be a prefix for the following instructions because the instructions following the impdep instructions would be decoded depending on the instruction.

16. Regarding claim 4, Chennupaty/Narayan/Leijten/Google disclose the processor of claim 3, wherein when detecting the Java impdep instruction, the subsequent instruction belongs to a different instruction set than the current instruction.

17. Regarding claim 13, Chennupaty/Narayan/Leijten/Google disclose the method of claim 12, wherein the prefix value is equal to a Java impdep instruction (see above regarding claim 3).

18. Regarding claim 15, Chennupaty/Narayan/Leijten/Google disclose he method of claim 9, wherein if the a Java wide prefix is detected, the first and second behaviors

comprise a first mode for decoding instructions of a first format and a second mode for decoding instructions of a second format (see above regarding claim 3).

19. Regarding claim 16, Chennupaty/Narayan/Leijten/Google disclose the method of claim 9, wherein if a Java impdep prefix is detected, the first and second behaviors comprise a first mode for decoding instructions of a first instruction set and a second mode for decoding instructions of a second instruction (see above regarding claim 3).

20. Regarding claim 27, Chennupaty/Narayan/Leijten/Google disclose the programmable device of claim 25, wherein the prefix is a Java impdep instruction (see above regarding claim 3).

21. Claims 5, 6, 14, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chennupaty (U.S. Patent No. 6,014,735) in view of Narayan (U.S. Patent No. 6,014,735) and in further view of Leijten and JVM (The JavaTM Virtual Machine Specification).

Regarding claim 5, Chennupaty/Narayan/Leijten disclose the processor of claim 1.

Chennupaty/Narayan/Leijten do not disclose the use of a Java wide instruction.

JVM teaches the use of a Java wide instruction for modifying a subsequent instruction by increasing the number of bytes used (wide instruction description).

One advantage of using a Java wide instruction is to be able to use larger indexes for loads and stores.

In light of this advantage, it would have been obvious for one of ordinary skill in the art at the time of the invention to use the Java wide instruction as a prefix in the combined invention of Chennupaty/Narayan/Leijten.

22. Regarding claim 6, Chennupaty/Narayan/Leijten /JVM discloses the processor of claim 1, wherein when detecting the Java wide instruction changes format of the subsequent instruction.

23. Regarding claim 14, Chennupaty/Narayan/Leijten/JVM discloses the method of claim 12, wherein the prefix value is equal to a Java wide instruction (see above regarding claim 5).

24. Regarding claim 26, Chennupaty/Narayan/Leijten /JVM discloses the programmable device of claim 25, wherein the prefix is a Java wide instruction (see above regarding claim 5).

25. Claims 17, 18, 19, 20, 23, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chennupaty (U.S. Patent No. 6,014,735) in view of Narayan (U.S. Patent No. 6,014,735) and in further view of Leijten and Nazomi (Nazomi Introduces First Universal Java Accelerator Chip for Mobile Wireless Applications).

26. Regarding claim 17, Chennupaty/Narayan/Leijten disclose a system, comprising: processor unit comprising: decode logic; and pre-decode logic associated with the decode logic; wherein the decode logic decodes a current instruction concurrently with the pre-decode logic determining if a subsequent instruction comprises a prefix in which case a program counter skips the prefix thereby precluding the decode logic from receiving the prefix and changes the decode logic operation during the decoding of the subsequent instruction (see above regarding claim 1).

Chennupaty/Narayan/Leijten do not expressly disclose that the processing unit is a co-processor coupled to a main processor.

Nazomi discloses the use of a Java co-processor on a cellular telephone (paragraph 1).

One advantage of using a Java co-processor on a cellular telephone is to speed up Java software execution (Nazomi, paragraph 5).

In light of this advantage, it would have been obvious for a person of ordinary skill in the art at the time of the invention to implement the processor of Chennupaty/Narayan/Leijten as a Java co-processor on a cellular telephone coupled to a main processor.

27. Regarding claim 18, Chennupaty/Narayan/Leijten/Nazomi discloses the system of claim 17, wherein concurrently with the decode logic decoding the current instruction, the pre-decode logic examines a predetermined number of subsequent bytes (see above regarding claim 7).

28. Regarding claim 19, Chennupaty/Narayan/Leijten/Nazomi discloses the system of claim 18, wherein the predetermined number is at least 5 (see above regarding claim 8).

29. Regarding claim 20, Chennupaty/Narayan/Leijten/Nazomi discloses the system of claim 18, wherein the predetermined number of subsequent bytes is compared to a prefix value (see above regarding claim 12).

30. Regarding claim 23, Chennupaty/Narayan/Leijten/Nazomi discloses the system of claim 17, wherein the instructions are of variable length (see above regarding claim 9).

31. Regarding claim 24, Chennupaty/Narayan/Leijten/Nazomi discloses the system of claim 17, wherein the system comprises a cellular telephone (see above regarding claim 18).

32. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chennupaty/Narayan/Leijten/Nazomi in view of Google.

33. For motivation and combination, see above regarding claim 3.

34. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chennupaty/Narayan/Leijten/Nazomi in view of JVM.

35. For motivation and combination, see above regarding claim 5.

Response to Arguments

36. Applicant's arguments filed 11 November 2008 have been fully considered but they are not persuasive.

37. Applicant states:

> The Examiner does assert, however, that Chennupaty discloses precluding the decode logic from receiving the prefix at fig. 4, ref. 420 and EN1, and col. 5, II. 59-65.

Although Chennupaty teaches “the decode logic causes a program counter to skip the prefix” and “precluding the decode logic from receiving the prefix”, it does not expressly disclose causing “a program counter to skip the prefix **thereby** …” Inherently, if a byte is skipped (as is shown in Chennupaty; see above regarding claim 1) the program counter must be incremented past that byte. The only deficiency in the Chennupaty reference is the causality between incrementing the program counter and precluding the logic from receiving the prefix.

38. Applicant states:

> Contrary to the Examiner's assertion, Leijten does not disclose incrementing a program counter to skip bytes that should not be decoded. Instead, paragraph 17 of Leijten teaches that a current instruction in a current memory line signals whether the program counter has to "skip padding" in a remaining part of the current memory line to advance to a start of the subsequent memory line. Stated in another way, Leijten teaches that when dealing with a particular memory line, a current instruction in that line can cause the program counter to skip padding in another part of that memory line, such that the next memory line is processed.

Examiner disagrees. Leijten clearly teaches skipping the program counter in order to pass over data which is unneeded (the padding). Clearly, the padding will not be decoded and will not be executed because it is merely padding (the entire point of padding is that it is useless data; in this case to fill space to the end of the cache line).

39. Applicant states:

> However, Leijten does not teach or even suggest that such skipping prevents decoding of the remainder of the memory line; it only teaches that such skipping prevents execution of the remainder of the memory line. In fact, given that entire memory lines are fetched at a time (para. 16, lines 3 and 5-6), and further given that the instruction that causes the skipping has already been decoded and executed, it is quite likely that the part of the memory line that is skipped has already been decoded and is ready for execution. Thus, because Leijten does not teach that its skipping prevents decoding (as required by claim 1), and further because Leijten actually suggests that decoding of the skipped portion has already taken place, the Examiner is mistaken when he states that Leijten teaches incrementing a program counter that prevents a decode logic from receiving a byte or prefix.

Examiner disagrees. Chennupaty teaches precluding data from being sent to the decode logic as stated above.

Conclusion

40. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JESSE R. MOLL whose telephone number is (571)272-2703. The examiner can normally be reached on M-F 10:00 am - 6:30 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alford Kindred can be reached on (571)272-4037. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jesse R Moll
Examiner
Art Unit 2181

/J. R. M./
Examiner, Art Unit 2181

/Niketa I. Patel/
Primary Examiner, Art Unit 2181